In the Claims:

1. (Previously presented) A method for building a complete three-dimensional ("3-D") model using an application neutral format, as an output that is readable and editable in a 3-D computer-aided-design ("CAD") software system, comprising:

building a plurality of features based on a feature class to give a plurality of built features, wherein the feature class comprises feature geometry, feature constraints, and feature dimensions, wherein the feature constraints are divided into two classes, and wherein one feature constraint class defines whether the constraint is to an edge or to a point and a definition of the edge or the point;

defining each built feature as a geometric representation of an individual feature type;

ordering the plurality of built features using geometry of up to six orthographic views of the built features, wherein the plurality of built features are ordered consistently with the CAD system's ordering expectations;

building a 3-D feature-based model based on the ordering of the plurality of built features to give a representation; and

storing the representation in an intermediate binary file format.

Claims 2 – 5. Cancelled.

6. (Previously Presented) The method of claim 1, wherein the intermediate binary file format comprises a geometry library and a feature library adapted to build the three-dimensional model.

7. (Previously Presented) The method of claim 6, wherein the geometry library comprises geometry classes for:

two-dimensional entities; three-dimensional entities- line; arc; elliptical arc; polyline; spline; face; points; and vectors.

8. (Original) The method of claim 7 further comprising copying data between at least one of the class's private data space and an address of the data specified from a calling function.

9. (Previously presented) The method of claim 8 further comprising, within each class, classifying the data as at least one of a following classification from a group consisting of:

fundamental data; and

derived data, wherein derived data is any additional information that may be used during the building of the 3-D model.

10. (Original) The method of claim 9 further comprising ensuring, by each of the classes, that any change made to the fundamental data via a function will update the derived data accordingly.

11. (Original) The method of claim 1 further comprising independently defining each feature via a three-dimensional coordinate system.

12. (Original) The method of claim 11, wherein the three-dimensional coordinate system contains the data necessary to detect at least one of a following element from a group consisting of:

a work plane;

a sketch plane; and

a face upon which a feature may need to be built.

13. (Original) The method of claim 12, wherein the data comprises at least one of a following element from a group consisting of:

plane vectors;

an origin of the plane; and

an elevation of the plane from a world origin.

Claims 14 – 18. Cancelled.

19. (Previously Presented) The method of claim 1, wherein the feature constraints are handled via a class that provides at least one of a following action from a group consisting of:

defining a constraint type, a constraint data value, and a constraint object; and

indicating if the constraint is to an edge or to a point, and a definition of the edge or the point, wherein the indicating is based on a constraint object type.

20. (Previously presented) The method of claim 1, wherein the intermediate binary file format may contain stored two-dimensional input views via a class, wherein the stored two-dimensional input are capable of being used to provide subsequent 3-D model validation.

21. (Currently amended) The method of claim 20, wherein each view class contains at least one of a following element from a group consisting of:

an array of two-dimensional entities; and

a three-dimensional coordinate system associated with the view.

22. (Previously Presented) The method of claim 1, further comprising transferring system specific data through an intermediate binary file based on the ordering of the built features.

23. (Cancelled)

24. (Previously Presented) The method of claim 1 further comprising transferring application specific data through an intermediate binary file based on the ordering of the built features.

25. (Cancelled)

26. (Previously Presented) The method of claim 1, wherein the intermediate binary file format is a binary file of individual features and metadata associated with each feature is created by serializing object data structures of individual features and associated metadata.

27. (Cancelled)

28. (Original) The method of claim 1, wherein the binary file format can be incrementally updated.

Claims 29 – 57. (Cancelled)

58. (Previously presented) A method for converting a two-dimensional drawing to a complete three-dimensional model, as an output that is readable and editable in a 3-D computer-aided-design ("CAD") software system, comprising:

(a) inputting the two-dimensional drawing;

(b) correcting errors associated with the two-dimensional drawing to give a corrected two dimensional drawing;

(c) using an automated feature detection system to create matched feature loops;

(d) performing a profile analysis and a feature analysis on the matched feature loops;

(e) producing an ordered list of three-dimensional features using geometry of up to six orthographic views of the three-dimensional features, wherein the ordered list of three-dimensional features is ordered consistently with the CAD system's ordering expectations;

(f) writing the ordered list of three-dimensional features to an intermediate binary file format;

(g) interfacing the binary file format to a binary file format that is specific for the CAD system; and

(h) producing a parametric feature-based three-dimensional model.

59. (Cancelled).

60. (Cancelled).

61. (Previously presented) The method of claim 58 further comprising back projecting the three-dimensional model to obtain drawing views associated with a three-dimensional model for the purpose of validating the three-dimensional model against the two-dimensional drawing.

62. (Previously Presented) The method of claim 61 further comprising overlaying the drawing views on top of the two-dimensional drawing views.

63. (Original) The method of claim 62 further comprising comparing the views.

64. (Cancelled)

65. (Previously presented) The method of claim 58, wherein:

step (a) comprises:

automatically filtering non-graphical entities, wherein the non-graphical entities include all dimension lines, centerlines, construction lines, hatching, text, title blocks, and borders, and

exploding any blocks in the drawing to accumulate indivisible geometric entities; and

step (b) comprises:

performing error checking on the drawing,

if errors are found, correcting the errors, and

automatically splitting entities in the drawing or in the corrected drawing corresponding to top, front and side views.

66. (Original) The method of claim 65 further comprising fixing a common origin for each view.

67. (Original) The method of claim 66 further comprising translating the entities to the common origin.

68. (Previously Presented) The method of claim 67 further comprising writing translated geometric entity data to classes.

Claims 69 – 70. (Cancelled)

71. (Previously presented) The method of claim 58, wherein step (c) comprises:

receiving the corrected two-dimensional drawing;

performing a subpart extraction of the corrected two dimensional drawing, wherein the subparts are join features including single features or combinations of features and wherein all subparts are extracted;

performing a subpart matching of the corrected two dimensional drawing;

extracting nested loops and circular loops;

matching the nested loops and circular loops; and

producing matched feature loops.

72. (Previously presented) The method of claim 58, wherein:

step (d) comprises:

receiving the matched feature loops,

performing a profile analysis on each loop match,

building feature subtrees, wherein each of the feature subtrees contains necessary data to create a 3-D feature,

setting a relative volume operation for each of the feature subtrees, and

building feature relations on the feature subtrees; and

step (e) comprises:

building a model tree based on the feature relations, and

producing a final feature tree based on the model tree to give the ordered list of three dimensional features.